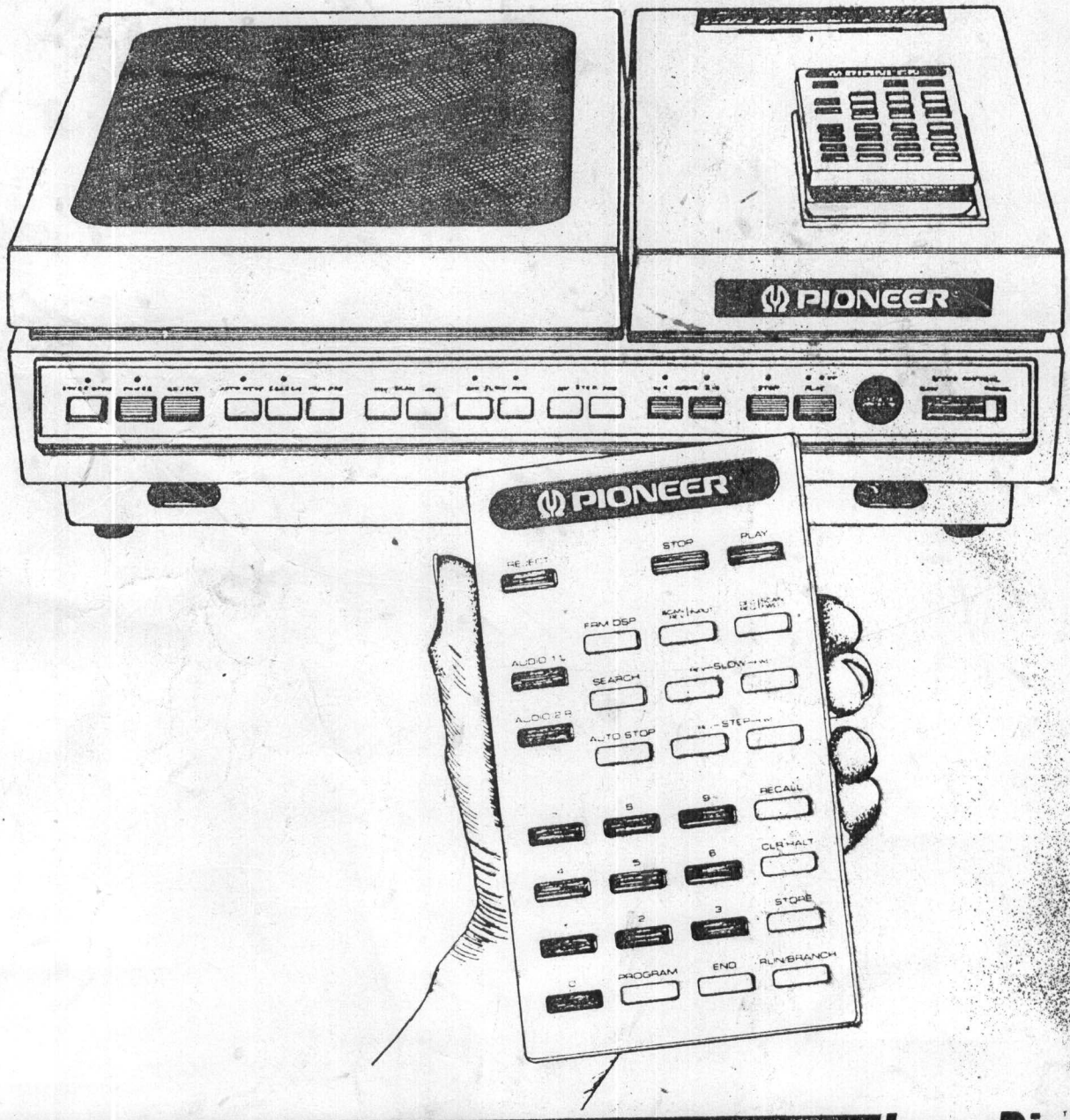
PIONEER

PIONEER PR7820 MODEL THREE LASERDISC™ PLAYER

PROGRAMMING REFERENCE GUIDE



LaserDisc™

TECHNICAL PUBLICATION

TP103

This document has
been scanned by

CelGen
Technologies

Redistribution of
this copy
permitted only at
no cost.
NOT FOR SALE

PREFACE

The Optical Videodisc Player, Model PR-7820-3, is the third in this series of videodisc players available from Pioneer Video, Inc. Although similar in many respects to previous models, the Model 3 Player incorporates many improvements, including an enriched instruction set, user definable function keys, increased program execution speed, and improved disc search time.

Users of the PR-7820-1 and PR-7820-2 Videodisc Players should note the following compatibility considerations:

- Digital programs developed for the Model 3 Player may not operate as intended on a Model 1 or Model 2 Player. The instruction set for the Model 3 Player contains many commands not recognized by previous player models.

- The Model 3 Player's microcomputer executes program instructions six times faster than previous models. Therefore, programs developed for Model 1 and Model 2 Players that contain timing loops or delays based on the microprocessor's execution speed should be examined to assess compatibility with the Model 3.

- To avoid erroneous entries from a Remote Control Unit when the player is operating under program control, only the commands HALT and REJECT will interrupt a program running on the Model 3 Player. Therefore, programmed discs which allow the user to enter other commands to interrupt program execution may not be compatible.

- The Model 1 External Computer Interface is upward compatible with that of the Model 3. The External Computer Interface for the Model 2 Player is not compatible with that of either the Model 1 or Model 3.

For all disc applications, it is recommended that the application, its program dumps, and any other features designed into the application (e.g., chapters, picture stops, etc.) be thoroughly tested. Features intended for use on several players should be carefully tested on each model player before being submitted to Pioneer Video, Inc. for disc mastering.

TABLE OF CONTENTS

CHAPTER I. INTRODUCTION

		PAGE
1.	SCOPE	1
2.	GENERAL	1
3.	VIDEODISC	2
	A. CAV Format	2
	B. CLV Format	3
4.	VIDEODISC PLAYER	3
5.	PROGRAM LOADING	3
	A. Program Dumps	4
	B. Remote control Unit	4
	C. Attached Host Computer	4
6.	OPERATING MODES	4
	A. Manual Mode	4
	B. Automatic Mode	5
	C. Programming Mode	6

CHAPTER II. PROGRAMMING INFORMATION

1.	GENERAL	7
2.	RANDOM ACCESS MEMORY	7
	A. Addressable Memory	7
	B. User Registers	8
3.	USER PROGRAM	8
	A. Arguments	9
	B. Commands	9
	C. Program Structure	10
4.	COMMAND DESCRIPTIONS	10
	A. Mode Control Commands	13
	1) PROGRAM - Programming Mode	14
	2) END - End Programming Mode	15
	3) RUN - Run Program	16
	B. Program Load Control Commands	17
	1) LOAD - Load Program From Disc	18
	2) PLOAD - Partial Load From Disc	20

TABLE OF CONTENTS (CONTINUED)

PAGE

C. Audio Control Commands 22

1) AUDIO 1/2 - Audio Track Control. 23
2) Axx - Set Audio Status 24

D. Video Control Commands 25

1) AUTOSTOP - Automatic Stop ... 26
2) DISPLAY - Frame Display 28
3) FSM - Field Step Mode 29
4) LPD - Landing Pad 31
5) PLAY - Play Motion Sequence at Normal Speed 32
6) SEARCH - Search to a Frame ... 34
7) SLOW FWD/REV - Slow Play Forward or Reverse 35
8) STEP FWD/REV - Step Single Frame Forward or Reverse 36
9) VOFF - Video Off 37
10) VON - Video On 38
11) WAIT - Delay Freeze Frame ... 39
12) WFW - White Flag Wait 40

E. Register Management Commands 41

1) ADD - Add to Register 0 42
2) ARG - Argument 43
3) CLK - Clock Read and Reset ... 44
4) COM - Compare Contents of Register 0 45
5) DEC REG - Decrement Register ... 46
6) DROP - Drop Low-Order Digit ... 48
7) GET - Set Value Into Register 0. 49
8) PUT - Transfer Value From Register 0 50
9) RECALL - Recall and Activate Register 51
10) RND - Generate "Random" Number . 52
11) STORE - Store Number in Active Register 54
12) SUB - Subtract Number From Register 0 55

F. Input Processing Commands 56

1) BIN - Button Input 57
2) DIN - Data Input 58
3) INPUT - Input Viewer Data ... 59

PAGE

4) FIN - Viewer Input With Function Keys 62
5) FTI - Viewer Input with Function Keys and Timeout ... 64
6) TINPUT - Viewer Input With Timeout 66

G. Program Execution Control Commands... 68

1) BRANCH - Transfer to Location... 69
2) BRF - Branch on Failure 70
3) HALT - Halt Program Control ... 71
4) JUMP - Jump to Subroutine ... 72
5) NE - No Entry 73
6) PAUSE - Pause in Execution ... 74
7) REJECT - Reject Videodisc ... 75

H. Non-Programmable Functions 76

CHAPTER III. THE EXTERNAL COMPUTER INTERFACE

1. GENERAL 77
2. USING THE EXTERNAL COMPUTER INTERFACE ... 77
3. COMMUNICATION PROTOCOL 78

A. PR-7820 Model 1 Compatibility Mode... 78
B. Extended Communications Mode 79

4. DATA FORMATS 79

A. Eight Bit Binary 79
B. Two ASCII Characters 79
C. One ASCII Character 80

5. HARDWARE CONSIDERATIONS 81

A. The Ready Line 81

1) PR-7820 Model 1 Compatibility Mode 81
2) PR-7820 Model 3 Extension of Ready Function 82
3) Serial Data Extraction From Ready 82
4) A Read-Confirm Generated From Ready 82

B. F8 Hardware/Software Monitoring ... 82

1) F8 Registers 83
2) F8 Input/Output Ports 84

TABLE OF CONTENTS (CONTINUED)

PAGE

6. EXTENDED COMMAND SET 85

A. Enable or Disable Character-Generator and Ready-As-Input-Strobe Flags ... 86
B. Enable or Disable Binary Output and Unique Input Flags 87
C. Transmit Frame Number 89
D. Transmit Status 90
E. Set Transmit Pointer 91
F. Increment or Decrement Transmit Pointer 92
G. Transmit Memory (Contents of RAM) ... 93
H. Transmit F8 Register 95
I. Transmit Contents of F8 Input Ports . 96
J. Transmit Contents of F8 Output Ports. 97
K. Transmit Program Address Pointer ... 98

CHAPTER IV. DATA ENTRY AND DISPLAY

1. GENERAL 99
2. CHARACTER DISPLAY 99
3. RCU PROGRAM ENTRY 100
4. RCU VIEWER RESPONSE 102

APPENDIX A. SUMMARY OF BASIC COMMANDS 104
APPENDIX B. SUMMARY OF COMPUTER COMMANDS 106
USER'S COMMENT FORM 107

CHAPTER I.

INTRODUCTION

1. SCOPE

This manual provides information to assist programmers in the design and development of application programs for the Pioneer Optical Videodisc Player, Model PR-7820-3, marketed by Pioneer Video, Inc., Montvale, New Jersey, and referred to in this manual as the Videodisc Player.

The videodisc player is a standalone system which, when connected to a television set or television monitor, comprises a self-contained video retrieval and display system. The primary emphasis of this manual is directed toward the standalone application; however, the videodisc player may also be connected to and controlled by a remote computer. Chapter III presents general interface and control information for users who may wish to attach the player to a computer. More detailed information, and assistance with specific computer-controlled applications, may be obtained from Pioneer Video, Inc.

2. GENERAL

The videodisc player is a highly flexible, user-programmable playback system that employs a laser to read television program material from a rotating videodisc. Information obtained from the videodisc is converted to signals capable of driving a standard NTSC television monitor or receiver.

Because the videodisc player is programmable, the exact sequence and display of information presented to the viewer can be predetermined by the program designer. Audio/visual messages can be developed for a wide variety of applications and tailored in unique ways. In addition, programs can be constructed which allow viewer-interaction with the displayed message. By entering numerical responses from a Remote Control Unit (RCU), the viewer may select particular topics for display; respond to questions and receive information appropriate to his response; and, in general, shape the presentation of the audio/visual message to his unique requirements. Properly formatted, the program-controlled videodisc brings outstanding flexibility to the presentation of audio/visual messages in industry, business, education, and entertainment.

3. VIDEODISC

The videodisc is the storage medium for the video and audio information processed by the videodisc player. It is a molded plastic disc with a 30 centimeter diameter, and with a capacity of up to 54,000 data tracks per side. The tracks are arranged on the disc in a continuous spiral, with information beginning at the inner circumference of the disc and proceeding to the outer.

Data on the videodisc exists in the form of microscopic depressions ("pits") in the surface of a disc side. This surface is coated with a thin aluminum layer which, in turn, is covered with a thin plastic coating to protect it during subsequent manufacturing processes. In practice the reflective surfaces of two disc sides are bonded together, back to back, to form a two-sided disc with the recorded information sandwiched between the two sides. The videodisc player's laser beam passes through the transparent plastic disc material of the outer disc surface and reads the recorded information from the raised reflective impressions ("bumps") within. The outer plastic disc surfaces also help to protect the reflective layer from mechanical damage.

Videodiscs may be recorded in either of two formats - Constant Angular Velocity (CAV) or Constant Linear Velocity (CLV).

A. CAV Format

A videodisc formatted for CAV is designed to rotate at a constant 1800 rpm. This format provides up to 30 minutes of television program material per side. Each of the 54,000 data tracks contains one complete television frame. Because the tracks are physically unique and regularly spaced, individual frames can be displayed in forward or reverse direction, at slow or normal speed. They may also be displayed step-by-step in either direction, or in single freeze-frame mode.

In addition to being physically distinct, each frame is logically identified by a unique frame number. Individual frames can thus be addressed by the player's internal logic and accessed by the laser. This capability makes possible many of the user-programmable features of the player, such as search for specified frame, branch between noncontiguous frames, and selected segment play.

B. CLV Format

A CLV formatted disc is designed to be read at a linear rate that is constant over the entire disc. To achieve this, the angular rotation rate of the disc is varied during playback from 1800 rpm at the innermost track to 600 rpm at the outermost. The number of frames per track varies, the quantity increasing with circumference. Although this format provides up to 60 minutes of television program material per side, discrete frames cannot be individually addressed. As a consequence, the previously mentioned CAV program control features are not available.*

4. VIDEODISC PLAYER

The videodisc player employs a low-power laser to read the data on the videodisc. Playback is achieved by directing a highly focussed laser beam onto the disc, recovering the reflected optical signal, and supplying it to a photodiode. The photodiode converts the optical data to electrical signals, which are further processed and ultimately converted to NTSC compatible video and VHF signals. These signals can drive either a television monitor or a conventional television receiver. Two channels of audio are separately available for connection to an optical user-supplied stereophonic sound system. The combined audio signals are also available as part of the RF modulated video signal for use with a standard NTSC television receiver.

An internal microcomputer controls all phases of videodisc player operation, processing both external command functions and internally-generated command, control, and status signals. The integration of a microcomputer into the videodisc player design makes possible the many play, search, and display functions of which the player is capable.

The microcomputer, in addition to exercising direct control of videodisc player operation, provides the user with one kilobyte (1024 bytes) of programmable Random Access Memory (RAM). This memory space is available for the storage of user-designed programs to control player operation. Programs stored in RAM are interpreted and controlled by a permanent interpreter program located in Erasable Programmable Read Only Memory (EPROM). No provision is provided to access the microcomputer instruction set directly.

5. PROGRAM LOADING

User programs may be loaded into RAM in one of three ways - from programs coded on the videodisc, manually from a Remote Control Unit, or under the control of an attached host computer.

---

*For the remainder of this manual, all references to videodiscs assume that the discs are CAV formatted.

A. Program Dumps

In addition to audio and visual information, a videodisc may be encoded with one or more "dumps." A dump is a computer program of 1022 bytes or less that is written on Audio Track 2 during the disc mastering process. Program dumps may be loaded into RAM from a videodisc. There can be multiple dumps on the same disc under user program control; each may overwrite 1022 bytes or less of the RAM contents.

B. Remote Control Unit

Computer programs may be written into RAM through use of the Remote Control Unit (RCU) supplied with each videodisc player. The RCU duplicates most of the front panel controls of the player, but contains additional keys for entering program instructions and data. The RCU permits a programmer to design and debug player control programs prior to their becoming finalized as program dumps on the videodisc. Programmers may also modify existing programs and tailor them to their unique requirements.

C. Attached Host Computer

The External Computer Interface (ECI) is a special connector at the rear of the videodisc player which permits interconnection of the player and a host computer. Aspects of videodisc player operation which can be accomplished through use of the RCU, including the loading of user programs into microcomputer RAM, may also be performed via the ECI. (Interface and control information is contained in Chapter III of this manual.)

6. OPERATING MODES

The videodisc player may be operated in any of three modes: Manual, Automatic, or Programming. Each mode is briefly described below.

A. Manual Mode

Manual Mode is initiated by mounting a disc, closing the player's protective cover, and pressing the PLAY button.* Assuming that the videodisc does not contain a program dump (or that Audio Track 2 is initially turned off, thereby inhibiting the loading of a program dump), the disc may be played from beginning to end without any further action by the viewer. The viewer may, however, vary the presentation of the material at any time through use of the front panel controls or the buttons on the RCU. (Refer to the Operator's Manual for a complete description of Manual Mode operation.)

---

*Some Model 3 Videodisc Players are equipped with a user-specified option that begins play of the videodisc as soon as the protective cover is closed and the power is turned on.

Manual Mode terminates under the following conditions:

- Videodisc Reject. Pressing the REJECT button causes videodisc rotation to slow to a stop, and the player to return to the PARK position.

- Automatic Mode Select. If the viewer presses the RUN button, the player enters Automatic Mode and begins executing the application program stored in RAM.

- Program Dump. If Audio Track 2 is on, and a disc containing one or more program dumps is started up from a previous REJECT, the player loads the initial program dump into RAM. When the load has completed, the player switches to Automatic Mode and begins executing the program.

- Programming Mode Select. If the viewer presses the PROGRAM button, any play in progress stops in a freeze-frame and the player enters Programming Mode.

Note: The viewer may press the STOP button to temporarily stop the video image in freeze frame; however, the player remains in Manual Mode, and action resumes when the PLAY button is pressed.

B. Automatic Mode

In Automatic Mode, operation of the videodisc player is controlled by the application program stored in RAM. This mode is automatically initiated by the successful loading of a program dump, or manually selected by pressing the RUN button on the RCU. If the viewer wishes to begin program execution at a specific memory location, he enters the location number prior to pushing the RUN button. If no location is specified, program execution begins at location 0000.

Viewer input during Automatic Mode is limited to the following:

- Audio Control. Both Audio Track 1 and Audio Track 2 are normally turned on when the player enters Automatic Mode. The viewer may at any time turn Audio Track 1 and Audio Track 2 on or off, undetected by the application program.

- Frame Display. The display of frame numbers is turned off when the player enters Automatic Mode. The viewer may turn this display on and off by pressing the FRM DSP button on the RCU, undetected by the application program.

- Viewer Response. If the application program has been written to allow viewer interaction, it will normally request this interaction by presenting a statement or question, followed by multiple choices. The viewer may, at that time, make the entry that corresponds to his choice. Subsequent operation of the program can be conditioned by that choice.

- Mode Termination. The viewer may terminate Automatic Mode by pressing either the HALT or the REJECT button. A HALT command returns the player to Manual Mode; however, the player may continue in motion until the STOP button is pressed, depending on the program command which was being executed just prior to the HALT. A REJECT button-press rejects the videodisc, as discussed previously for Manual Mode.

C. Programming Mode

Programming Mode provides the user with direct access to both the "addressable" RAM memory and user registers of the videodisc player's microcomputer.

Programming Mode is initiated by pressing the PROGRAM button on the RCU. The user may then enter commands, arguments, and register data to construct a program that will subsequently control the operation of the player in Automatic Mode. During Programming Mode, the television set displays the numerical values and command mnemonics entered from the RCU. In addition, the programmer may review any selected memory location by entering the location number, and pressing PROGRAM. Each succeeding PROGRAM button-press advances the display by one location.

Programming Mode is normally terminated by pressing the END button, thus returning the player to Manual Mode. The application program that was created will assume control of the player when the RUN button is pressed.

Note: The user may terminate Programming Mode by pressing REJECT; this action will not cause the REJECT command to be stored in the program.

The remaining chapters of this manual describe the commands that are available to the programmer, and provide instructions for entering them into the microcomputer RAM.

CHAPTER II.

PROGRAMMING INFORMATION

1. GENERAL

This chapter provides the user with the information necessary to develop application programs for use with the videodisc player. It contains a brief description of the programmable memory and registers, describes the essential characteristics of a user program, and presents a comprehensive discussion of program commands.

Throughout the remainder of this manual, extensive reference is made to hexadecimal values. Hex values are denoted by a leading asterisk, e.g., *F2. Although not standard notation this method has been adopted because the video display employs the asterisk convention to display many of the commands that are entered into RAM during Programming Mode.

2. RANDOM ACCESS MEMORY

The videodisc player's F8 microcomputer contains the basic operating system of the player and also provides one kilobyte (1024 bytes) of Random Access Memory (RAM). The RAM performs a dual function, serving as both addressable memory and user registers. This coexistence allows RAM to be used both for program storage and as a group of 512 registers that the program can reference. Because memory and registers may overlay each other, the programmer must understand how each is structured and addressed.

A. Addressable Memory

Each RAM memory location is equivalent to one eight-bit byte. Memory locations begin at address 0 and continue to address 1023. Locations 1022 and 1023 are not normally used for program storage, but are reserved for use as Register 0.

Program data, i.e., arguments and commands, are stored in coded format. Each argument digit and each command require one byte, and thus occupy one memory location. For example, the value 1536 stored within a program as a command argument occupies four bytes and has the hexadecimal format *0F*AF*4F*6F.

B. User Registers

Each user register occupies two bytes of RAM (two memory locations). There are 512 possible registers, addressed 0 through 511. Register numbers begin at the high end of memory and proceed downward: Register 0 occupies memory locations 1022 and 1023, while Register 511 occupies memory locations 0 and 1.

Note: The two memory locations corresponding to a particular register number can be computed as follows:

Mem Loc = 1022 - (2 x Reg. No.)
The register will occupy Mem Loc and Mem Loc+1, with the register high order byte value stored in Mem Loc and the low order byte in Mem Loc+1.

Register data are stored in a 16-bit binary format. For example, the value 1536, which occupies four bytes of memory, can be stored in a two-byte register, and has the hexadecimal value *0600.

Since registers are 16 bits long, they can contain values of 0 through 65536. Negative numbers, or larger positive numbers are taken modulo 65536.

Of the 512 available registers, only one is "active" at any given time. With the Model 1, a register had to be active before its contents could be used or modified by the program.

Register 1 is set active by a RUN command entered from the RCU. Any register can be designated as active by specifying it as the argument of a RECALL command. In addition, successive registers are automatically activated whenever one of the following commands is executed, even if the command does not use the register: AUTOSTOP, SEARCH, STORE, and RECALL.

Note: The videodisc player's F8 microcomputer contains additional registers which may be accessed by an attached computer. These registers are specifically identified in text as F8 registers. All other references to registers are to the user registers contained in RAM.

3. USER PROGRAM

A user program for the videodisc player consists of a series of commands that, when stored in RAM and executed by the microcomputer, causes the player to operate in a predefined way. The following is a brief description of arguments and commands, and gives an overview of a program structure. Detailed descriptions of arguments and commands are contained in later sections of this chapter, as well as in Chapter III.

A. Arguments

Arguments specify the numerical parameters required by their associated commands. An argument may specify an integer data value, a frame number, a memory location, or a register number. These arguments may denote time delays or register content data such as the acceptable range of numeric values entered by the viewer in response to input commands.

Some commands do not require arguments; others require arguments; for still others, an argument is optional. In the case of optional arguments, there is usually a default value, or an implied argument in the active register.

Each numeric digit of an argument is internally represented as a two digit hexadecimal value. When stored in memory, each of these two digit hex values occupies one memory location. When the programmer enters numeric values from the RCU, the video display shows the numeric value of the button-press, and not its hexadecimal representation.

Any number of digits may be entered immediately prior to a command, but only the low-order five digits are interpreted as the argument. The numerical value of these low-order digits is taken as modulo 65536.

B. Commands

Commands are the functional instructions that the programmer may use to develop an application program. Many of the commands are direct counterparts of buttons on the RCU (e.g., AUDIO 1, FRM DSP, etc.). When executed in Automatic Mode, they cause the corresponding operations to be performed by the player. Other commands are used for controlling the program itself, directing the path of execution, managing registers, etc.

Each command is internally represented by a two digit hex value. (Refer to Appendix A.) Many of these values can be entered directly from the RCU as a single button press; others must be entered as two separate hex digits, following the procedure described in Chapter IV. When stored in memory, each command occupies one location (one byte). When the programmer enters commands from the RCU, the video display shows some of the commands as hex values, and others as command mnemonics.

C. Program Structure

A user program, when stored in memory, resembles a continuous string of characters. The string is processed by a program interpreter contained in the player's microcomputer. Beginning at the location specified or implied by the RUN command, each element of the string is inspected for a command code. When the command is located, an internal command processor performs the required function. Characters preceding the command code are treated as the command argument. Some arguments have implied or default values; in such cases there will not be any preceding argument in the character string.

As an example, the commands 1000 SEARCH 2000 AUTOSTOP are internally represented as *0F*3F*3F*3F*F7*8F*3F*3F*3F*F3. Starting with the first byte, *0F, the argument is accumulated while the character string is being scanned for a command code. In this example, the SEARCH command code, *F7, is detected. The SEARCH command processor, using the currently accumulated argument (*0F*3F*3F*3F), instructs the player to position the laser at frame 1000. When the frame has been located, the next command is processed.

In general, succeeding commands are not processed until the function specified by the "current" command has been completed. The PLAY command is the only exception. A PLAY command instructs the videodisc player to begin playing audio/video material, and to continue until terminated by the program. Program execution continues while the play is in progress.

4. COMMAND DESCRIPTIONS

This section provides a detailed description of each command in the videodisc player's instruction set, except those related to the use of an attached computer. Computer information and commands are presented separately in Chapter III.

The program commands are grouped by function:

Mode Control
Program Load Control
Audio Control
Video Control
Register Management
Input Processing
Program Execution Control

C. Program Structure

A user program, when stored in memory, resembles a continuous string of characters. The string is processed by a program interpreter contained in the player's microcomputer. Beginning at the location specified or implied by the RUN command, each element of the string is inspected for a command code. When the command is located, an internal command processor performs the required function. Characters preceding the command code are treated as the command argument. Some arguments have implied or default values; in such cases there will not be any preceding argument in the character string.

As an example, the commands 1000 SEARCH 2000 AUTOSTOP are internally represented as *0F*3F*3F*3F*F7*8F*3F*3F*3F*F3. Starting with the first byte, *0F, the argument is accumulated while the character string is being scanned for a command code. In this example, the SEARCH command code, *F7, is detected. The SEARCH command processor, using the currently accumulated argument (*0F*3F*3F*3F), instructs the player to position the laser at frame 1000. When the frame has been located, the next command is processed.

In general, succeeding commands are not processed until the function specified by the "current" command has been completed. The PLAY command is the only exception. A PLAY command instructs the videodisc player to begin playing audio/video material, and to continue until terminated by the program. Program execution continues while the play is in progress.

4. COMMAND DESCRIPTIONS

This section provides a detailed description of each command in the videodisc player's instruction set, except those related to the use of an attached computer. Computer information and commands are presented separately in Chapter III.

The program commands are grouped by function:

Mode Control
Program Load Control
Audio Control
Video Control
Register Management
Input Processing
Program Execution Control

Within each function group, individual commands are presented in accordance with the standard format shown below:

#. **ABBREVIATION** - Command Title

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM

The first line of the description contains an abbreviation of the command and the command title, e.g. AUTOSTOP - Automatic Stop. For clarity, the abbreviation is used throughout the description and example that follow.

The second line contains a tabulation of command programming information:

ARGUMENT — If an argument is required for a particular command,the table indicates the argument type, i.e., frame number, register number or integer. If the argument is optional, it is enclosed in parentheses.

CMD MNM — Following the argument is a command nmemonic which the programmer should use when coding a program to be submitted to Pioneer Video, Inc.

HEX — The hexadecimal value that follows the mnemonic identifies the command to the program interpreter. During Programming Mode, the programmer may enter this value into memory by following the hex-entry procedure described in Chapter IV.

EQUIVALENT RCU BUTTON RCU — For many commands, the hexadecimal value of the command may be entered directly as a single button-press, rather than as a two character value. The table indicates what button, if any, corresponds to the command being described.

DSPLY MNM — Commands entered from the RCU during Programming Mode are displayed on the video screen at the time of entry. Some commands are represented by one or two-letter mnemonics; others are shown as hex values. The table indicates the displayed characters for the command under discussion.

A. Mode Control Commands

Mode Control commands are those the user enters manually from the RCU to initiate and terminate Programming Mode and to start executing a program stored in RAM.

The commands described in this section are:

1. PROGRAM
2. END
3. RUN

The commands PROGRAM and END should not be included in a user program, even though it is possible to enter the hexadecimal codes for each, using the hex-entry techniques described in Chapter IV.

The RUN/BRANCH button is used to enter the BRANCH command when in Programming Mode.

1) **PROGRAM** - Programming Mode

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Address)	None	DF	PROGRAM	None

The PROGRAM command is entered from the RCU to initiate Programming Mode. Following entry of this command, new data can be entered into RAM, or existing data can be reviewed. A specific memory location may be immediately accessed by entering the desired address, followed by pressing the PROGRAM button. If no address is specified, location 0000 is assumed. Argument values greater than 1023 are taken modulo 1024.

Program entries are stored in successive memory locations. Each instruction and each digit of an argument occupy one memory location. The video screen displays the active memory address and the contents of that address. Subsequent operation of the PROGRAM button advances the display address by one location.

Previously stored information can be edited or updated during a program review by entering a new command or character to replace the one being examined. The user should take care not to overlay needed commands or leave irrelevant data.

Normal termination of Programming Mode is acomplished by entering an END command. (A REJECT command will also terminate Programming Mode; the videodisc will stop rotating and the player will return to the PARK position).

2) **END** - End Program Mode

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	None	EF	END	None

The END command is manually input by the programmer to terminate Programming Mode and to return the player to Manual Mode.

3) **RUN** - Run Program

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Address)	None	CF	RUN	None

Programs stored in RAM may be initiated manually or under program control.

Manual initiation of a stored program requires that the player be in Manual Mode, and is accomplished by pressing the RUN button on the RCU. This action causes the microcomputer to begin executing the program at a specified location, or at location 0000 if no argument is specified. A manually entered RUN turns on Audio Tracks 1 and 2, turns off the frame-number display, sets Register 1 active, and transfers the player from Manual Mode to Automatic Mode.

If several separate programs (actually these are subprograms) are contained in RAM, one can initiate another by BRANCHing to its starting address. (Refer to discussion of BRANCH command.) The BRANCH command does not alter the existing status of Audio Tracks 1 and 2, and has no effect on the frame-number display or the active register.

B. Program Load Control Commands

The two Program Load commands described in this section are used in conjunction with program dumps on the videodisc. In general, they specify the number of data bytes to be loaded into the player's RAM.

The commands are:

1. LOAD
2. PLOAD

1) **LOAD** - Load Program From Disc

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	L	CC		*CC

The LOAD command is used in conjunction with a dump leader tone signal on the videodisc to load a 1022 byte program dump from the disc to the player's RAM. Note that only 1022 bytes, not 1024, are loaded. The remaining two bytes (addresses 1023 and 1024), which correspond to Register 0, are not contained in the program dump. Register 0 is therefore not overlayed by a dump, and may be used for passing data between different programs.

There are no prohibited byte values in a dump. Any of the hex codes contained in the player's instruction set may be loaded into RAM via a program dump.

The program dump is physically located on Audio Track 2. When the LOAD command is executed, valid program data must be detected from Audio Track 2 in order for the load to be successful. (It is not necessary for Audio Track 2 to be turned on, except in the case where the dump is to be loaded immediately following the initial start-up of the player.) If the load is successful, 1022 bytes of data are loaded, and program execution begins at location 0000.

Note: If no dump leader tone signal is detected, the player may default to Manual Mode. If the leader tone is detected, but the load cannot be successfully accomplished, the player may freeze-frame at the point where the load is supposed to begin, and then revert to Manual Mode.

Load data are recorded at 625 bytes per second and span approximately 50 frames of Audio Track 2. The combined dump leader tone and dump data may occupy 150 or more frames of Audio Track 2. Information regarding the recommended positioning and spacing of multiple program dumps may be obtained from Pioneer Video, Inc.

Example: The dump leader tone signal starts before frame 450, and the program dump starts after frame 450. The associated code would be:

```
450 SEARCH
    LOAD
```

2) **PLOAD** - Partial Load From Disc

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	PLD	0C		*0C

Similar to the LOAD command, the PLOAD command is used with a dump leader tone signal on the disc to load program data into RAM. Unlike the LOAD command which is used to access a full 1022-byte dump, the PLOAD command is used to load dumps of 1022 bytes or less. There are no prohibited byte values.

When a PLOAD command is executed, valid data must be detected on Audio Track 2. (Audio Track 2 does not need to be on.) Then 1024-N bytes of data (where N is the contents of Register 0) are loaded. The load begins at memory location 0000 and ends at location 1023-N. The remaining memory locations/ register contents (i.e., from location 1024-N through 1023) are not changed by the PLOAD command. This feature expands the ability to pass data from one program dump to another.

The default for an unsuccessful PLOAD is the same as for an unsuccessful LOAD. The number of disc tracks reserved for a PLOAD during disc mastering is the same as for a LOAD, even though the actual dump content may be less than 1022 bytes.

Caution: Do not use a value of N that is less than 2 or greater than 1023.

Example: The currently operating program calls a new dump from disc, where the dump leader tone signal starts before frame 450, the dump data start after frame 450, and the user wishes to overlay the first 500 bytes of the current program:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0600	450 SEARCH	Search to frame 450.
0604	524 GET	Place 524 in R0.
0608	PLOAD	Load 500 bytes (1024 - 524) from Audio Track 2 into RAM. (Load memory locations 0000 through 0499; locations 0500 through 1023 remain unchanged.)

C. Audio Control Commands

The following commands provide program control of the audio information contained on a videodisc:

1) AUDIO 1
2) AUDIO 2
3) Axx (where x can be I, F, N, or T)

1) **AUDIO 1/2** - Audio Track Control

ARGUMENT	CODE MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Integer)	A1	F4	AUDIO 1	A1
(Integer)	A2	FC	AUDIO 2	A2

AUDIO 1 controls Audio Track 1, and AUDIO 2 controls Audio Track 2. When the command is preceded by an even argument (i.e., 0, 2, 4...), audio for that particular track is turned off; an odd argument (i.e., 1, 3, 5...) turns audio on. If there is no preceding argument, the command acts as a toggle switch to turn audio on or off, depending on its previous state.

Note: The RUN command turns on both audio tracks. During Automatic Mode, the viewer may manually toggle either track at any time, undetected by the program.

Example: Play from frame 1000 to frame 3000 with Audio Track 1 off and Audio Track 2 on; from frame 3000 to frame 5000 with Audio Track 1 on and Audio Track 2 off.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0013	0 AUDIO 1	Turn off Audio Track 1.
0015	1 AUDIO 2	Turn on Audio Track 2.
0017	1000 SEARCH	Search to frame 1000.
0022	3000 AUTOSTOP	Play to frame 3000.
0027	AUDIO 1	Toggle Audio Track 1 on.
0028	AUDIO 2	Toggle Audio Track 2 off.
0029	5000 AUTOSTOP	Play to frame 5000 and freeze frame.

2) **Axx** - Set Audio Status

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	AFF	A0		*A0
	AFN	A1		*A1
	ANF	A2		*A2
	ANN	A3		*A3
	AFT	A4		*A4
	AFI	A5		*A5
	ANT	A6		*A6
	ANI	A7		*A7
	ATF	A8		*A8
	ATN	A9		*A9
	AIF	AA		*AA
	AIN	AB		*AB
	ATT	AC		*AC
	ATI	AD		*AD
	AIT	AE		*AE

The audio status commands, coded *A0 through *AE, allow the programmer to set Audio Track 1 and Audio Track 2 to any one of fifteen possible combinations through use of a single command. Audio tracks may be turned on, turned off, toggled, or ignored (left in the same state). The following table indicates the audio status specified by each command. Note the use of T for toggle, F for off, N for on, and I for ignore.

CMD	EFFECT ON AUDIO TRACKS
AFF	Turn off 2, Turn off 1.
AFN	Turn off 2, Turn on 1.
ANF	Turn on 2, Turn off 1.
ANN	Turn on 2, Turn on 1.
AFT	Turn off 2, Toggle 1.
AFI	Turn off 2, Ignore 1.
ANT	Turn on 2, Toggle 1.
ANI	Turn of 2, Ignore 1.
ATF	Toggle 2, Turn off 1.
ATN	Toggle 2, Turn on 1.
AIF	Ignore 2, Turn off 1.
AIN	Ignore 2, Turn on 1.
ATT	Toggle 2, Toggle 1.
ATI	Toggle 2, Ignore 1.
AIT	Ignore 2, Toggle 1.

D. Video Control Commands

The following commands control the selection, sequencing, and display of video information contained on the videodisc:

1)	AUTOSTOP	7)	SLOW FWD/REV
2)	DISPLAY	8)	STEP FWD/REV
3)	FSM	9)	VOFF
4)	LPD	10)	VON
5)	PLAY	11)	WAIT
6)	SEARCH	12)	WFW

1) **AUTOSTOP** - Automatic Stop

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Frame Number)	AS	F3	AUTOSTOP	AS

The AUTOSTOP command begins normal-speed play of a videodisc, starting at the current frame number and ending in freeze-frame at the frame specified by the argument. If no argument is specified, a default frame number is taken from the active register.

The AUTOSTOP command always causes the active register to be deactivated and the next register to be activated, whether or not an argument is specified.

Note: If the frame number associated with the AUTOSTOP command is less than the current frame number, the player will not play in reverse; it will search to the target frame and may, in the process, display other frames. It is strongly recommended that the programmer avoid the deliberate use of this phenomenon in the design of the programmed application.

Example: Play the first 1000 frames of a sequence, delay five seconds and play the next 1500 frames.

Assume:

R50 = 2000
R51 = 1000
R52 = 4500

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0010	50 RECALL	Activate Register 50.
0013	SEARCH	Search to frame 2000 (the frame number stored in the active register) and activate Register 51.
0014	3000 AUTOSTOP	Play to frame 3000; activate Register 52. (The value in Register 51 is not used.)
0019	50 STOP	Display frame 3000 for 5 seconds.
0022	AUTOSTOP	Play to frame 4500 and freeze frame; activate Register 53.

2) **DISPLAY** - Frame Display

ARGUMENT	CODE MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Integer)	DI	F1	FRM DSP	DI

The DISPLAY command controls the display of the five-character frame number. An even argument (i.e., 0, 2, 4...) turns the frame number display off; an odd argument (i.e., 1, 3, 5...) turns it on. If used without an argument, DISPLAY acts as a toggle switch to turn the frame number on or off, depending on its previous state.

Note: The RUN command turns the frame display off. During Automatic Mode, the viewer may toggle the frame display at any time, undetected by the program.

Example: Ensure that frame numbers will be displayed when the first video sequence begins playing:

Assume R0 = 1000 and R1 = 2000.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0048	0 RECALL	Activate R0.
0050	SEARCH	Search to frame 1000 (as stored in the active register) and activate R1.
0051	1 DISPLAY	Turn on the frame display.
0053	AUTOSTOP	Play to frame 2000 (as stored in the active register) with the display on; activate R2.
0054	DISPLAY	Turn off the frame display.
0055	3000 AUTOSTOP	Play to frame 3000 with the frame display off; activate R3.

3) **FSM** - Field Step Mode

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	FSM	06		*06

The FSM command may be followed by STEP FWD or STEP REV command to step the display image forward or back one field. Two adjacent fields continue to be displayed as one video frame. The player continues in Field Step Mode until a STOP command is encountered in the code.

Note: If a numeric argument is encountered while FSM is active, the player may revert to normal freeze frame mode. However, it is strongly recommended that use of the numeric digit to terminate FSM be avoided in the design of programmed applications.

Example: Following a manually-entered 5 RUN command, search to frame 100, then sequentially step back 55 fields:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	STEP REV	Step back one video field.
0001	ARG	ARG becomes the number of the active register (R0).
0002	DEC REG	Decrement the active register. If the contents > 0, go to the next instruction at location 0003. If the contents = 0, skip one BRANCH and go to location 0004.
0003	BRANCH	Go to location 0000.
0004	HALT	Terminate program control; return to Manual Mode.
0005	1 DISPLAY	This is the first command executed after "5 RUN" is entered. Turn on the frame display.
0007	100 SEARCH	Search to frame 100.
0011	0 RECALL	Activate Register 0.
0013	55 GET	Place 55 in Register 0.
0016	FSM	Enter Field Step Mode.
0017	BRANCH	Go to location 0000. (The instructions in locations 0000 through 0003 comprise a loop that cycles 55 times, stepping the player in reverse for 55 fields before the loop is exited and the HALT at MEM LOC 0004 is performed.

4) **LPD** - Landing Pad

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	LPD	01		*01

The LPD command modifies the number of times the player attempts to locate a frame following an unsuccessful SEARCH. A SEARCH attempts to locate a specified frame and, if unsuccessful, makes two retries at the next two higher frame numbers. The LPD command extends the number of retries from 0 through 63, depending on the argument.

If a SEARCH and the retries are unsuccessful, the SEARCH ends on a nearby, but indeterminate frame, and the next program instruction is executed. For recovery procedures, see BRF.

If the argument value (N) for LPD is 63 or less, the player retries the SEARCH the number of times specified; on each new attempt the next higher frame number is sought. If the argument value is greater than 127, it is interpreted as modulo 128. For such a case, the original frame number will be retried N-128 times.

Caution: Do not use arguments of 64 through 127, or any argument value greater than 191 with the LPD command. Use of these values will inhibit subsequent SEARCH functions.

The value established by the LPD command remains in effect throughout the program unless another LPD command is executed or the disc is REJECTED. The REJECT command stops disc rotation, returns the player to the PARK position, and restores the number of SEARCH retries equal to two.

Example: The command 10 LPD specifies retry ten times at ten ascending adjacent frames. The command 138 LPD means retry ten times at the same frame. A 0 LPD means make no retries.

5) **PLAY** - Play Motion Sequence at Normal Speed

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	P	FD		P.

The PLAY command instructs the videodisc player to begin playing the disc at normal speed, starting at the current frame and continuing for an indeterminate time. When the PLAY command is entered into memory from the RCU, it must be entered as *FD. (Direct operation of the PLAY button in Programming Mode conditions the player to accept the next two button presses as a hexadecimal command value. Refer to Chapter IV.)

Program execution continues while the PLAY is in progress. Play is terminated when one of the following commands is executed:

AUTOSTOP	SLOW REV
REJECT	STEP FWD
SEARCH	STEP REV
SLOW FWD	STOP

The HALT command ends program execution, but does not terminate PLAY.

The PLAY command permits program execution to continue while the disc is being played, and thus permits motion sequences to be displayed on a time basis, rather than a frame basis.

Example: Begin playing a disc at frame 1000, and continue for 3.1 seconds. If interrupted during that time by a 0 entered from the RCU, continue for an additional 3.1 seconds; however, if interrupted by a 1 through 9, display frames 2000 through 3000 before starting again at frame 1000. If there is no interruption in the last 3.1 seconds, display frame 3000 for 1 second and start over.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	1000 SEARCH	Search to frame 1000.
0005	PLAY	Begin playing the videodisc at frame 1000.
0006	31 TINPUT	Accept viewer input of 0 or "other" (1 through 9) with a timeout after 3.1 seconds: If input is 0, go to location 0009. If input is "other", skip one BRANCH and go to location 0011. If timeout occurs go to location 0022.
0009	6 BRANCH	Go to location 0006. Assuming no further viewer input, the disc will continue to play for 3.1 seconds before the timeout condition is satisfied.
0011	2000 SEARCH	Search to frame 2000.
0016	2100 AUTOSTOP	Play to frame 2100.
0021	BRANCH	Go to location 0000.
0022	3000 SEARCH	Search to frame 3000.
0027	10 WAIT	Display frame 3000 for 1 second.
0030	0 BRANCH	Go to location 0000.

6) **SEARCH** - Search to a Frame

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Frame Number)	SC	F7	SEARCH	SC

The SEARCH command instructs the player to position the laser at the frame number given in the argument. If no frame number is provided, the frame number in the active register is used. Video is squelched during the SEARCH; at the end of the SEARCH the image is displayed in freeze-frame. Completion of the SEARCH activates the next register.

If the player is unable to locate the target frame, it makes two retries at the next two higher frame numbers. If these retries are unsuccessful, the SEARCH ends on a nearby, but indeterminate frame, and the next program instruction is executed.

Program control of the player following an unsuccessful SEARCH can be directed by the BRF command. The programmer may expand the number of entries attempted following an unsuccessful SEARCH through use of a preceding LPD command.

Example: The command 1500 SEARCH directs the player to search to frame 1500, stop, and display the image in freeze-frame.

7) **SLOW FWD/SLOW REV** - Slow Play Forward or Reverse.

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Frame Number	SLF	F2	SLOW FWD	LF
Frame Number	SLR	FA	SLOW REV	LR

The SLOW FWD/SLOW REV commands display video in forward or reverse slow motion until the specified frame number is reached. Audio is suppressed during play, and play terminates in a freeze-frame. Play speeds range from 0 to 30 frames per second, depending on the setting of the SPEED CONTROL slide on the front panel of the player.

Note: If the SPEED CONTROL is moved to the full-left or SLOW position, slow play will not proceed until it is moved back to some intermediate position.

Caution: The use of a greater-than-current frame number with a SLOW REV command, or a less-than-current frame number with a SLOW FWD command will cause the player to search to the target frame; it may, in the process display other frames. The programmer should avoid the deliberate use of this phenomenon. An argument is required. The contents of the active register are not used as a default for the argument in these commands.

Example: Beginning at frame 26000, play backward in slow motion for 1000 frames.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0300	26000 SEARCH	Search to frame 26000.
0306	25000 SLOW REV	Play backward to frame 25000 and freeze-frame.

8) **STEP FWD/STEP REV** - Step Single Frame Forward Reverse

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	SF	F6	STEP FWD	SF
	SR	FE	STEP REV	SR

The STEP FWD/STEP REV commands cause the videodisc player to step forward or backward one frame for each execution of the command. These commands can be used in a programmed loop to display a series of still frames, or to provide programmed slow-motion effects.

Example: Beginning at the current frame, display the next 10 frames as a series of 8-second stills:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	20 RECALL	Activate Register 20.
0003	10 STORE	Place the value 10 in Register 20.
0006	STEP FWD	Step to the next higher frame.
0007	80 WAIT	Display the frame for 8 seconds.
0010	20 DEC REG	Decrement Register 20.
		If R20 > 0, go to location 0013.
		If R20 = 0, skip one BRANCH and go to location 0015.
0013	6 BRANCH	Go to location 0006.
0015	HALT	Terminate the program.

9) **VOFF** - Video Off

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	VFF	1C		*1C

The VOFF command turns off the video image normally seen during freeze-frame, leaving a blank screen. The freeze-frame image will be turned back on by execution of a VON command. In addition, execution of any one of the "motion play" commands will turn the display back on while they are executing. These commands are: PLAY, AUTOSTOP, SLOW FWD, and SLOW REV.

Note: If VOFF is active during a SEARCH/AUTOSTOP sequence, the video will be turned off during the search, turned on during the motion segment, and turned off when the AUTOSTOP reaches its target frame.

An example of use of VOFF is illustrated in the description of the VON command.

10) **VON** - Video On

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	VON	1B		*1B

The VON command turns on the freeze-frame video previously turned off by VOFF.

Note: The image on some television sets will roll when a freeze-frame image is turned back on.

Example: Flash video off and on.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	1000 SEARCH	Search to frame 1000.
0005	VOFF	Turn video off.
0006	10 WAIT	Delay 1 second.
0009	VON	Turn video on.
0010	10 WAIT	Delay 1 second.
0013	5 BRANCH	Loop again.

11) **WAIT** - Delay Freeze Frame

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	W	FB	STOP	W.

The WAIT command causes a freeze-frame pause for a period of time defined by the integer argument. The integer specifies the time in tenths of a second.

Example: A programmed freeze-frame for 5.2 seconds is coded as 52 WAIT.

The maximum freeze-frame delay available is 1000 seconds (10,000 tenth seconds or 10000 WAIT).

Note: If a PLAY command was active, the WAIT command will first cause the motion to stop in a freeze frame; then it will wait for the designated time before executing the next program command. If no argument is specified, the motion sequence (if any) will be stopped in a freeze frame, and the program will move to the next command immediately. (compare with the PAUSE command.)

12) **WFW** - White Flag Wait

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	WFW	1A		*1A

The WFW command permits synchronization of program commands with the White Flag. Execution of the WFW command requires one cycle if the field does not have the white flag on, and two cycles if the flag is on. This condition usually allows the next instruction byte to be interpreted on a White Flag field.

Example: Begin Field Step Mode on the White Flag field of a freeze-frame:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0100	1000 SEARCH	Search to frame 1000.
0105	WFW	Wait for the White Flag.
0106	FSM	Enter FSM at a frame boundary.

E. Register Management Commands

The following commands provide program control of the user registers and their contents:

1)	ADD	7)	GET
2)	ARG	8)	PUT
3)	CLK	9)	RECALL
4)	COM	10)	RND
5)	DEC REG	11)	STORE
6)	DROP	12)	SUB

1) **ADD** - Add to Register 0

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	ADD	02		*02

The ADD command adds the value contained in its argument to the contents of Register 0, then stores the sum into R0. The result of ADD is taken as Modulo 65536.

Example: Add 50 to the frame number in Register 0 and search to the new frame. Assume R0 = 1000.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0015	50 ADD	Add 50 to the value in R0 and store the sum (1050) back into R0.
0018	0 ARG	The value in R0 becomes an argument.
0020	SEARCH	The player searches to frame 1050.

2) **ARG** - Argument

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM.
(Register Number)	ARG	0A		*0A

ARG acts as a variable and takes on a numerical value which is used as an argument by the command which follows it. If ARG is preceded by an argument, ARG takes on a value that is equal to the active register number, but not its contents.

Examples: In the following list of examples, assume:

R0 = 13
R8 = 10
R10 = 208
R3 = 8 and is active

	CODE	EFFECT
o	10 ARG BRANCH	Go to location 0208.
o	8 ARG ARG BRANCH	Go to location 0208.
o	ARG BRANCH	Go to location 0003, R3 is active.
o	10 ARG GET	208 is placed in R0.
o	8 ARG PUT	13 is placed in R10.
o	ARG ARG BRANCH	Go to location 0008.
o	ARG ARG ARG BRANCH	Go to location 0010.
o	0 ARG RECALL	R13 becomes active.

3) **CLK** - Clock Read and Reset

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	CLK	16		*16

A tenth-of-a-second interval timer is accessible by the CLK command whenever the videodisc player is in Automatic Mode. This timer accumulates counts up to a value of 6553.5 seconds before beginning again at zero. Execution of the CLK command causes the "current" value of the timer to be placed in Register 2; sets the timer to zero; then restarts the timer.

Example: Determine the elapsed time for a search of 4,000 frames:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	1 SEARCH	Search to frame 1.
0002	CLK	Place the current time in R2, initialize, and restart the timer.
0003	4000 SEARCH	Search to frame 4000.
0009	CLK	GET the elapsed time for the search and store it in R2.

4) **COM** - Compare Contents of Register 0

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	COM	04	-	*04

The value of the argument for COM is compared to the content of R0. If the content of R0 is greater than the argument, the next sequential instruction is executed. If the value in R0 equals the argument, execution continues immediately after the next BRANCH command. If the content of R0 is less than the argument, execution continues immediately after the second BRANCH.

Example: Compare contents of R10 and R0.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0020	10 ARG	ARG becomes the value stored in Register 10.
0023	COM	Compare the value of ARG with the value stored in R0: If R10 < R0, go to location 0024. If R0 = R10, skip one BRANCH and go to location 0027. If R10 > R0, skip two BRANCHes and go to location 0030.
0024	33 BRANCH	Go to location 0033.
0027	39 BRANCH	Go to location 0039.
0030	45 BRANCH	Go to location 0045.

5) **DEC REG** - Decrement Register

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Register Number	DR	F0	DEC REG	DR

DEC REG command subtracts one from the contents of the register specified in the argument. If the result is greater than 0, the next command is executed; if it is equal to 0, all succeeding commands are skipped until one BRANCH is passed; the next command following the passed BRANCH is then executed. There are no negative numbers. A zero decremented remains a zero. An argument is required.

Note: A DEC REG command does not update the active register.

Example: Assuming that Registers 21 through 30 contain different frame numbers, the following code will play five scenes and then halt:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0050	20 RECALL	Activate Register 20 to contain the loop counter.
0503	5 STORE	Set the loop counter to 5; activate the next register (R21).
0505	SEARCH	Search to the frame number contained in the active register; activate the next register (i.e., R22).
0506	AUTOSTOP	Play to the frame number contained in the active register; activate the next register.
0507	20 DEC REG	Decrement R20 - the loop counter. If the contents > 0, go to location 0510. If the contents = 0, skip one BRANCH and go to location 0514.
0510	505 BRANCH	Transfer to location 0505.
0514	HALT	Terminate program control.

6) **DROP** - Drop Low Order Digit

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	DRP	1D		*1D

The DROP command deletes the low-order digit from the value specified by its argument (i.e., divides by 10 with no remainder). The new value is used as an argument by the next command.

Example: The following example changes the contents of R0 from 123 to 127:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0014	123 GET	Put 123 into R0.
0018	0 ARG	ARG becomes the value 123 taken from R0.
0020	DROP	The value of the argument becomes 12.
0021	7 GET	Numerical 7 is concatenated to the end of 12, and the new argument value 127 is stored in R0.

7) **GET** - Set Value Into Register 0

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	GET	08		*08

The GET command sets Register 0 to the value specified by its argument.

Example: The code 10 GET will store the value 10 in Register 10.

8) **PUT** - Transfer Value From Register 0

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Register Number	PUT	09		*09

The PUT command copies the contents of Register 0 into the register specified by the argument.

Examples:

	CODE	EFFECT
o	10 PUT	Copy the content of R0 into R10.
o	ARG PUT	Copy the contents of R0 into the active register.

9) **RECALL** - Recall and Activate Register

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Register Number)	RC	7F	RECALL	RC

The RECALL command activates the register specified by the argument. If no register number is specified, one of the following occurs:

- o If the active register number has not been updated since the last RECALL command, RECALL activates the next register.

- o If the active register number has been updated since the last RECALL command, a RECALL reactivates the already-active register, - i.e., the active register does not change.

Note: The following commands update the active register: SEARCH, AUTOSTOP, and STORE.

Example:

ARGUMENT AND COMMAND	DESCRIPTION
10 RECALL	Activates R10
RECALL	Activates R11
300 SEARCH	Activates R12
RECALL	Activates R12

10) **RND** - Generate "Random" Number

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	RND	05		*05

RND acts as variable, and takes on a pseudo-random value from 0 through 255 each time it is encountered. That value acts as an argument for the next instruction in memory.

Example: Beginning at location 4, display frame 1000 in some instances, frame 3000 in others:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0004	RND	RND becomes a random value, 0 through 225.
0005	GET	Place that random value in Register 0.
0006	0 ARG	ARG becomes equal to the contents of Register 0 - the random value.
0008	DROP	Drop the low-order digit from ARG - the random value. Determine the value of the low-order digit of the random number as follows:
0009	0 SUB	Concatenate 0 with the value of ARG; subtract results from contents of Register 0; store difference (i.e., the low-order digit) back into Register 0.
0011	5 COMP	Is the low-order digit equal to 5? If R0 > 5, go to next location (0013). If RO = 5, skip one BRANCH to location 0016. If R0 = < 5, skip two BRANCHes to location 0019.
0013	25 BRANCH	Go to location 0025 if the digit was 6 to 9.
0016	25 BRANCH	Go to location 0025 if the digit was 5.
0019	1000 SEARCH	Search to frame 1000 if the digit was 0 to 4.
0024	BRANCH	Return to location 0000.
0025	3000 BRANCH	Search to frame 3000 if the digit was 5 to 9.
0026	BRANCH	Return to location 0000.

11) **STORE** - Store Number in Active Register

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Integer)	ST	F5	STORE	ST

The STORE command places a value in the active register. The value stored is the number contained in the argument. If there is no argument, the current frame number is stored. Completion of the STORE command activates the next register.

Note: Argument values greater than 65535 are taken as Modulo 65536.

Example: Place the current frame number in Register 20 and the value 5000 in Register 21:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0010	20 RECALL	Activate Register 20.
0013	STORE	Store the current frame number in Register 20 and activate Register 21.
0014	5000 STORE	Store 5000 in Register 21 and activate Register 22.

12) **SUB** - Subtract Number from Register 0.

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	SUB	03		*03

The SUB command subtracts the value contained in its argument from the value in Register 0, and stores the difference in R0. Execution of a SUB command may yield a negative result. The actual value of that result is determined by subtracting the unsigned result value from 65,536. For example, if R0 contains a value of 10, then 12 SUB subtracts 12 from 10 and stores 65,534 (65,536-2) in R0.

Example: Subtract 50 from the frame number in R0 and search to the new frame. Assume R0 = 1200.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0015	50 SUB	Subtract 50 from 1200 in R0 and store the result in R0.
0018	0 ARG	ARG is the value in R0.
0010	SEARCH	Search to frame 1150.

F. Input Processing Commands

The commands described in this section are used to process viewer inputs entered from the RCU while a program is running. The subsequent path of program execution can be conditioned by the viewer's input, thereby permitting a wide range of interactivity between the viewer and the videodisc player.

The commands are:

1)	BIN	4)	FIN
2)	DIN	5)	FTI
3)	INPUT	6)	TINPUT

1) **BIN** - Button Input

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	BIN	17		*17

Each command and each numeric digit which is input to the videodisc player is input as a code having a decimal value in the range of 0 through 255. When BIN is encountered it takes on the decimal value of the code that is being externally entered to the player. This value is used as an argument by the command that follows BIN.

Example: An executing program loops until any button is pressed.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0005	255 GET	Place 255 in Register 0.
0009	BIN	BIN becomes the decimal equivalent of the hexadecimal value of a button press.
0010	COM	Compare the value of BIN with the contents of R0. If 255 > BIN, go to location, 0011. If BIN = 255, skip one BRANCH and go to location 0014. If 255 < BIN, skip two BRANCHes and go to location 0016.
0011	18 BRANCH	Go to location 0018.
0014	9 BRANCH	Return to location 0009.
0016	9 BRANCH	Return to location 0009.
0018	HALT	Terminate the program.

2) **DIN** - Data Input

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	DIN	1E		*1E

DIN can be used with any of the input commands to extend the range of non-numeric inputs to include the values 10 through 19. DIN takes on a numerical value which is used as an argument by the command which follows it. DIN can be used after any type of input command (INPUT, TINPUT, FIN, and FTI), and takes the value of 0 through 20, depending on the RCU button pushed. The numeric buttons give DIN the corresponding values 0 through 9. The reassigned RCU button-values for 10 through 19 are shown in Chapter IV.

Example: Following a button press between 0 and 9, place the associated numerical value in Register 0.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	1 INPUT	Accept viewer input and establish button 0 as the "valid" response. If input is 0, go to location 0002. If input is 1 through 9, skip one BRANCH and go to 0004.
0002	4 BRANCH	Go to location 0004; input was "0".
0004	DIN	DIN takes the numerical value (0-9) of the button push.
0005	GET	Using DIN as the argument, place the numerical value of the button press into R0.

3) **INPUT** - Input Viewer Data

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	IN	F8	INPUT	IN

The INPUT command allows the processing of a numeric entry by the viewer, usually made in response to a question or "menu" displayed on the video screen. The entry may be any single RCU button-press in the range 0 through 9. The digit entered becomes the index to a list of succeeding BRANCH commands, and directs the flow of the program.

The argument coded by the programmer is normally a single digit N. High-order digits, if present in the argument, are ignored. The argument N establishes the range (button-press 0 through N-1) of "expected" viewer input. Any numeric button press outside the expected range is treated as a single category, "other."

The following table shows the relationship between argument values, "expected" input and "other" input.

ARGUMENT VALUE (N)	RANGE OF "EXPECTED" NUMERIC INPUT	RANGE OF "OTHER" NUMERIC INPUT
1	0 only	1 through 9
2	0 or 1	2 through 9
3	0 through 2	3 through 9
4	0 through 3	4 through 9
5	0 through 4	5 through 9
6	0 through 5	6 through 9
7	0 through 6	7 through 9
8	0 through 7	8 and 9
9	0 through 8	9 only

NOTE: An argument value of 0 is treated the same as an argument value of 9.

For example, if N = 3, three numeric responses (0, 1 and 2) are expected; any other numeric entry falls into the category "other."

If the viewer enters a numeric digit that is within the expected range, that digit specifies the quantity of subsequent BRANCH instructions to be skipped in program execution. If the viewer's response is in the "other" range, the number of BRANCHes skipped is equal to N.

When coding an INPUT for three expected entries, there should be three groups of commands following the INPUT statement, each group ending with a BRANCH command.

Note: Some previous versions of the videodisc player require a non-significant leading zero in the first address following an INPUT command. Although this zero is not required for the PR-7820 Model 3, the user may wish to program it in order to provide program compatibility with earlier models. Note that the meaning of certain commands is changed when their arguments are specified as zero; i.e., SEARCH has a meaning different than 0 SEARCH, and STORE is different than 0 STORE.

Example: The following sequence of commands will cause a search to frame 1000 and will accept viewer input for two expected entries and one "other."

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0100	1000 SEARCH	Search to frame 1000.
0105	2 INPUT	Allow the viewer input of 0, 1, or "other": If 0, go to location 0107. If 1, skip one BRANCH and go to location 0119. If "other" (2-9), skip two BRANCHes and go to location 0131.
0107	1500 SEARCH	Search to frame 1500 (in response to a "0").
0112	55 STOP	Show frame 1500 for 5.5 seconds.
0115	139 BRANCH	Go to location 0139.
0119	2000 SEARCH	Search to frame 2000 (in response to a "1").
0124	55 STOP	Show frame 2000 for 5.5 seconds.
0127	139 BRANCH	Go to location 0139.
0131	2500 SEARCH	Search to frame 2500 (in response to a "2" through "9").
0136	55 STOP	Show frame 2500 for 5.5 seconds.
0139	.....	Next instruction.

4) **FIN** - Viewer Input with Function Keys

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	FIN	18		*18

The FIN command operates in a manner which is similar to the INPUT command, but with the following exceptions:

o Viewer response is not limited to numeric keys 0 through 9. Values of 10 through 19 can be input via the following redefined RCU buttons:

BUTTON	EQUIVALENT NUMERICAL VALUE
AUTOSTOP	10
STEP REV	11
STEP FWD	12
SEARCH	13
SLOW REV	14
SLOW FWD	15
SCAN REV	16
SCAN FWD	17
STOP	18
PLAY	19

o Similar to the INPUT command, the integer argument is normally a single digit N; high order digits are disregarded. The digit N is used to process inputs in the range of 0 through 9. If one of the above redefined function buttons is pressed by the viewer, the next N+1 BRANCH instructions are skipped as in the TINPUT command. The value of the function button is made available to the program by the DIN command.

Example: Ignore all inputs except 15 (SLOW FWD).

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	1 FIN	Allow viewer input of 0, "other", or the function keys.
		If 0 is input, go to location 0002.
		If the entry is 1-9 (i.e., "other"), skip one BRANCH and go to location 0003.
		If a function key is pushed, skip two BRANCHes, go to location 0004.
0002	BRANCH	Return to location 0000.
0003	BRANCH	Return to location 0000.
0004	DIN	Convert the button press to a numeric value (0 to 19).
0005	GET	Place that value in Register 0.
0006	15 COM	Was the button-push equivalent to 15?
		If 15 < R0, go to location 0009.
		If R0 = 15, skip one BRANCH and go to location 0010.
		If 15 > R0, skip two BRANCHes and go to location 0013.
0009	BRANCH	Return to location 0000.
0010	14 BRANCH	Go to location 0014.
0013	BRANCH	Return to location 0000.
0014	.....	

5) **FTI** - Viewer Input with Function Keys and Timeout.

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	FTI	19		*19

The FTI command is similar to the FIN command in that it permits the entry of values from 0 through 19. However, it also recognizes a timeout condition, assigning to it a numerical value of 20. The timeout duration is specified by the argument, as described for a TINPUT command. The maximum timeout is 10,000 tenth-seconds.

Example: For a viewer input of 0 through 19, play from frame 100 to frame 200; if no RCU button is pressed for 10.1 seconds, display frame 300:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0050	100 SEARCH	Search to frame 100.
0054	101 FTI	Accept viewer input of 0, "other" (1-9), or function button (10-19), with timeout after 10.1 seconds. If input is 0, go to location 58. If "other" (1-9), skip one BRANCH and go to location 0061. If function button (10-19) is pressed, or timeout occurs ("button" 20), skip two BRANCHes and go to location 0064.
0058	78 BRANCH	Input was 0; go to location 0078.
0061	78 BRANCH	Input was 1-9, go to location 0078.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0064	DIN	DIN becomes the value of the button pressed (10-19), or timeout value of 20.
0065	GET	Place the value of the button press or timeout in Register 0.
0066	19 COMP	Compare the contents of R0 to 19. If 19 < R0 (i.e., if timeout), go to location 0069. If R0 = 19, skip one BRANCH and go to location 0072. If 19 > R0, skip two BRANCHes and go to location 0075.
0069	83 BRANCH	Go to location 0083.
0072	78 BRANCH	Go to location 0078.
0075	78 BRANCH	Go to location 0078.
0078	200 AUTOSTOP	Play to frame 200.
0082	HALT	Terminate program.
0083	300 SEARCH	Search to frame 300.
0087	HALT	Terminate program.

6) **TINPUT** - Viewer Input with Timeout

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	TIN	*0E		*0E

The TINPUT command allows the viewer to enter a numeric digit (0 through 9) from the RCU during Automatic Mode. The digit entry becomes an index to a list of succeeding BRANCH commands, and directs the flow of the program. If the viewer does not respond within a predetermined time, execution resumes at a timeout-specified location. The maximum timeout is 10,000 tenth-seconds.

The TINPUT command operates in a manner similar to the INPUT command with the following exception; the low-order digit means the same as for INPUT, but the entire integer argument is also interpreted as a tenth-of-a-second timeout value. This value establishes a condition that forces the program to continue if no input is received within the specified timeout period. If the argument of the command is N, and if a timeout occurs, N+1 BRANCH instructions are skipped in execution.

As with the INPUT command, there should be one BRANCH instruction present for each expected input, and an additional BRANCH for "other" inputs. There should be an additional BRANCH for the timeout condition.

Example: Display frame 100 and wait for viewer input. If the viewer enters a 1, show frame 300. If the viewer enters any other numeric digit, or does not respond within 10.2 seconds, show frame 200.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	100 SEARCH	Search to frame 100.
0004	102 TINPUT	Allow viewer input of 0, 1, and "other", with timeout after 10.2 seconds. If input is 0, go to next location (0008). If input is 1, skip one BRANCH to location 0011. If any other input, skip two BRANCHes and go to location 0014. If no response within 10.2 seconds, skip 3 BRANCHes and go to location 0017.
0008	20 BRANCH	Go to location 0020.
0011	25 BRANCH	Go to location 0025.
0014	20 BRANCH	Go to location 0020.
0017	20 BRANCH	Go to location 0020.
0020	200 SEARCH	Search to frame 200 in response to an entry of 0, "other", or timeout; then stop.
0024	HALT	Terminate program.
0025	300 SEARCH	Search to frame 300 in response to an entry of 1; then stop.
0029	HALT	Terminate program.

G. Program Execution Control Commands

The following commands are used to control and alter the order in which program steps are executed:

1) BRANCH
2) BRF
3) HALT
4) JUMP
5) NE
6) PAUSE
7) REJECT

1) **BRANCH** - Transfer to Location

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Address)	BR	CF	BRANCH	BR

The BRANCH command alters the sequential flow of execution by directing the program to continue execution at the memory location specified by the integer argument. If no argument is present, location 0000 is assumed.

BRANCHes are generally used under the following conditions:

- For conditional branching or loop control following a DEC REG or compare command.
- For path selection, where a series of BRANCH commands follows an input command (INPUT, TINPUT, etc.).
- An unconditional branch to cause immediate transfer of control to another location.

Examples:

	CODE	EFFECT
o	BRANCH	Transfer control to location 0000.
o	256 BRANCH	Transfer control to location 00256.
o	2 ARG BRANCH	Transfer control to the location specified in Register 2.

2) **BRF** - Branch on Failure

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Address	BRF	07		*07

The BRF command directs the order of program execution following an unsuccessful SEARCH. If the SEARCH is not accomplished within the specified number of retries (refer to SEARCH, and LPD commands), the BRF command transfers program execution to the location specified in its argument. If the SEARCH is successful, the BRF command is ignored.

Example: Following an unsuccessful SEARCH, branch to a memory location different than if the SEARCH were successful.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0024	100 SEARCH	Search to frame 100.
		If unsuccessful, try frames 101 and 102.
0028	75 BRF	If the initial try, or the two retries are successful, go to location 0031.
		If these tries are unsuccessful, BRANCH to location 0075.
0031	.....	Instruction if SEARCH is successful.
0075	.....	Instruction if SEARCH is unsuccessful.

3) **HALT** - Halt Program Command

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	H	BF	HALT	H.

The HALT command terminates program control of the player. It takes the player out of Automatic Mode and places it in Manual Mode. (Programs need not necessarily contain a HALT instruction, as they may be designed to execute continuously.)

Note: If a programmed HALT is encountered while a PLAY is in progress, program control terminates, but the videodisc continues to play in Manual Mode.

Example: Play from the current frame to frame 10,000; display frame 10,000 in freeze-frame; and place the player in Manual Mode.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0065	10000 AUTOSTOP	Play from the current frame to frame 10,000 and freeze-frame.
0071	HALT	Place the player in Manual Mode.

4) **JUMP** - Jump to Subroutine

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	JMP	0B		*0B

The JUMP command causes the program logic to transfer execution to the memory location specified by the argument. It also stores, in Register 1, the address of the memory location immediately following the JUMP command. This address may be used as a subroutine return.

Example: From memory location 56, JUMP to a subroutine at location 350, then return to the location that follows the subroutine call.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0056	350 JUMP	Place memory location 60 in R1 and cause program execution to continue at location 0350.
0060	.....	
.	.	
.	.	
.	.	
0350	.....	The subroutine at location 0350.
.	.	
.	.	
.	.	
0398	1 ARG	ARG takes the value 60, stored in R1.
0400	BRANCH	Using ARG as the argument, transfer execution to location 0060.

5) **NE** - No Entry

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	NE	FF		*FF

NE instructs the computer to "do nothing", but to execute the next command. Although transparent to program execution, each NE physically occupies one memory location. NE's may be entered from the RCU in order to blank unwanted commands which are part of an already loaded program, or to reserve space for future program modifications. Program-controlled NE's can be used to eliminate existing commands by overwriting them.

Note: Leading (non-significant) zeros in command arguments achieve the same effect as NE's. Execution of NE or of non-significant zeros requires one program cycle.

6) **PAUSE** - Pause in Execution

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	PAU	0D		*0D

The PAUSE command causes program execution to be suspended for the duration of time specified by the argument. The integer value of the argument is expressed in tenths of a second. The videodisc player continues to play during this time. Therefore, the command may be used to program a timed PLAY sequence.

Example: Play the first 13.5 seconds of material after frame 1000:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0100	1000 SEARCH	Find frame 1000.
0105	PLAY	Begin play.
0106	135 PAUSE	Suspend program execution for 13.5 seconds.
0110	WAIT	Stop at the present time and freeze the current frame.

The maximum time is 10,000 tenth-seconds.

7) **REJECT** - Reject Videodisc

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	RJ	F9		RJ

The REJECT command terminates program execution, stops disc rotation, and returns the player to the PARK position. It causes a 1 to be stored in Register 0. REJECT is a legitimate programming instruction. It may be entered into the player's RAM only in hex format (see Chapter IV). When executed, it will cause the program to abruptly terminate by rejecting the videodisc.

In addition, the following effects occur as a result of a REJECT:

- Frame Display is turned off
- Player enters Manual Mode
- Landing Pad is set equal to 2
- VON is enabled
- AUDIO 2 is turned on if the player door is opened; AUDIO 1 is unaffected.
- The Character Generator Disable Flag is reset (Chapter III)
- The Unique Input Flag is reset (Chapter III)

H. Non-Programmable Functions

Three of the commands discussed previously, PROGRAM, END, and RUN, are manual control commands that are not normally included in a stored program.

Two additional commands, SCAN FWD (*F0) and SCAN REV (*F8) may be manually input from the RCU while the videodisc player is operating in Manual Mode. In Manual Mode, these commands will cause the player's laser to scan rapidly in a forward or reverse direction until the corresponding button is released. If, however, these commands codes are encountered in Automatic Mode, they will be interpreted as DEC REG (*F0) and INPUT (*F8) commands.

CHAPTER III

THE EXTERNAL COMPUTER INTERFACE

1. GENERAL

The PR-7820-3 Videodisc Player operates as a standalone device in Manual or Automatic Modes as described in this publication. In addition, it may be attached to an external computer via the External Computer Interface (ECI) located at the rear of the player. Specifications for Model 1-compatible use of the ECI may be found in Pioneer Video Publication number TP101. This chapter describes the extended function and commands available through use of the Model 3's ECI. A careful study of TP101 is recommended before attempting to use the ECI.

2. USING THE EXTERNAL COMPUTER INTERFACE

Commands entered via the External Computer Interface are interpreted exactly as if they had come from the RCU. With INT/EXT low, the ENTER line operates as a button-press, with the duration of the press being significant for the SCAN FWD/REV commands.

The External Computer Interface differs from the RCU in that only 30 different bytes may be directly generated from the RCU. However, all 256 bytes (*00 through *FF) may be transmitted as data on the ECI data lines. Of those, six are treated specifically, and cannot be entered casually as part of program register data:

- The *FF is equivalent to not putting any data on the lines (all lines float high), and is the byte normally read by the PR-7820 when no entry is being made (ENTER high, no button-press).
- The *00 byte is converted to *FF by the microcode, thus giving the appearance of no command entry.
- The END command (*EF) terminates Programming Mode and, as is the case of an END entered from the RCU, is not entered into memory.
- The PLAY command (*FD) is translated into a hex prefix command during Programming Mode. It is treated as a PLAY command in Manual Mode, but not in Automatic Mode.

- The PROGRAM command (*DF) is treated as a request to skip a byte in memory, and to not insert a new byte at that memory location. This operation is identical to use of the PROGRAM button on the RCU.

- The REJECT command (*F9) causes the disc to be rejected, even when in Programming Mode.

- In addition to the 6 bytes above, bytes 3Ø to 39, 4Øto 4F, BØ to B7, and CØ to CF are treated as ASCII characters, and may not be casually entered as part of a program.

The SCAN FWD (*F0) and SCAN REV (*F8) commands cause the player's laser to rapidly scan the disc in a forward or reverse direction. These commands do not use target frame addresses; the laser continues to scan for the duration of the "button press." It may therefore be difficult to use these commands effectively with some interfaces. Note that the SCAN commands move the laser at approximately 300 frames per second when issued from STOP mode, and at approximately 3000 frames per second when issued from the PLAY mode.

Caution: An application of the SCAN FWD/REV commands that moves the player past recorded material on the videodisc may cause the disc to begin rotating in reverse.

3. COMMUNICATION PROTOCOL

Communication protocol compatibility between the Model 1 and Model 3 versions of the PR-7820 has been provided. This compatibility permits existing customer-developed host-attached applications to be immediately transferable from the Model 1 player to the Model 3 player, whether or not a UEI has been used. However, the Model 3 player provides an alternative (user selectable) communication protocol which permits new host-attached applications to capitalize on the faster processing capabilities of the Model 3.

A. PR-7820 Model 1 Compatibility Mode.

The Model 3 player requires that a command be sampled once; however, in Model 1 Compatibility Mode if a command is sent for multiple sample periods the transmission is processed as only one instance of that command. Commands may be followed either by intervals of no data, or immediately by a different command. Each ECI or RCU command may be held on the internal microprocessor's data bus for as long as the user wishes.

If two sequential commands are identical, they should be separated by an *FF to distinguish them from one single command that is "held" for an extended duration. For example, a 1000 SEARCH might be input as a 1, 0, *FF, 0, *FF, 0 SEARCH.

B. Extended Communications Mode

By enabling the appropriate flags, the Model 3 player may be set to operate in Extended Communications Mode.

When the player is in this state, each command sample is considered a unique instance of the command. It may be followed directly by any other command, another identical command, or any number of No-Data (*FF or *ØØ) conditions. A command that is held on the data bus for multiple sample periods is considered as multiple instances of that command.

4. DATA FORMATS

Data sent to the ECI may be formatted in any one of three formats: eight-bit binary; two ASCII characters; or one ASCII character.

A. Eight-Bit Binary

In this format, the bit-pattern must be equivalent to that represented by the hex value for that command.

Example:

PLAY: *FD = 111111Ø1
Numeric 2: *8F = 10001111
PLOAD: *ØC = 00001100

B. Two ASCII Characters

Data may be formatted as two ASCII characters, Ø through 9 and A through F. When it is more convenient to transmit only ASCII (7 bits plus parity), two characters for each byte may be transmitted as shown above. The first hex character is the ASCII character representing the high-order 4-bit nibble of the command byte. The second ASCII character represents the low-order 4-bit nibble. For example: A RECALL (*7F) may be entered as an ASCII 7 followed by an ASCII F.

The bit patterns for the ASCII characters are shown in the table below. The high order bit may be a 1 or Ø, depending upon the user's selection of parity - shown as a P in this table. Both are acceptable to the ECI.

CHARACTER	HEX CODE		BINARY
	PARITY = 1	PARITY = Ø	
0	30	B0	P0110000
1	31	B1	P0110001
2	32	B2	P0110010
3	33	B3	P0110011
4	34	B4	P0110100
5	35	B5	P0110101
6	36	B6	P0110110
7	37	B7	P0110111
8	38	B8	P0111000
9	39	B9	P0111001
A	41	C1	P1000001
B	42	C2	P1000010
C	43	C3	P1000011
D	44	C4	P1000100
E	45	C5	P1000101
F	46	C6	P1000110
@	40	C0	P1000000

In the event that a nibble is somehow lost in the process of data communication between the player and the attached computer, the at-sign (@) cleans out any previously-saved-nibble, such that the next ASCII hex character is considered to be the high-order nibble.

C. One ASCII Character

When the bit-pattern of a byte (including parity) exactly matches a command's hex value, it may be formatted as one ASCII character and will be acceptable by the ECI. This mode is specifically provided to facilitate the entry of the TRANSMIT commands. It is particularly useful when the player is required to execute the command immediately, rather than to store it in the player's RAM.

Example: The TRANSMIT STATUS command is represented in hex as a *D4. If the parity bit is set at EVEN or "1", this command is equivalent to an ASCII T. The other TRANSMIT commands are equivalent to ASCII P through Z, and various special symbols. Most of the player's standard commands can also be entered this way, if the parity is FORCED-1.

5. HARDWARE CONSIDERATIONS

Data bytes are transmitted bit-serially on the player's Ready Line in a standard asynchronous start-stop telecommunication format. Thus, the bits are sent low-order bit first at 9600 bits per second - one start bit, eight data bits, and at least two stop bits. This transmission requires 1.14 milliseconds (11 x 104 microseconds).

The bytes transmitted may be ASCII characters, or any eight-bit byte of *00 to *FF. The transmission is completed within approximately 6 milliseconds of reading the command. TRANSMIT commands may send either one or two characters in this time period, depending upon the specific command and the setting of the Binary-Output Mode flag. A TRANSMIT command may transmit its data even before the Enter Line is raised.

Commands may be read at 60 per second. The Ready Line is normally High at a TTL level of approximately +4 Volts (unloaded). The start bit, and data zeros cause the line to drop to Low (TTL approximately zero Volts).

Although the TTL voltage levels are not standard RS232 voltage levels (± 5 Volts), most available integrated circuit RS232 receiver chips have a "fail-safe" mode of operation that will receive these TTL levels. Many industry-standard UART chips satisfactorily receive the data directly. Standard RS232 ports clocked for 9600 baud, full duplex, asynchronous start-stop telecommunication will likely accept the Ready Line <u>if inverted</u>. The PR-7820 Ready Line is driven by a 74LS02.

A. The Ready Line

1) PR-7820 Model 1 Compatibility Mode.

As with the Model 1, the Ready pulse occurs once each cycle. However, data are read <u>during</u> the Ready pulse, at least 2 microseconds inside pulse boundaries, not between ready pulses as on the Model 1. The pulse length is normally 16 microseconds, but varies up to 20 microseconds if the character generator is on. The cycle is considered to be the interval between vertical sync pulses, i.e., twice per revolution of the disc (17 milliseconds). When the vertical sync is absent or missed, the period may be from 20 to 30 milliseconds. This usually occurs during PARK, Startup, REJECT, when the disc is not being read by the laser, or occasionally during a SEARCH or SCAN.

The Ready pulse is absent when the player is "BUSY." BUSY conditions for the Model 3 are essentially the same as for the Model 1 player, except that pulses do appear during non-timeout INPUT commands.

2) PR-7820 Model 3 Extension of Ready Function.

a) The Model 3 extends the function of the Ready pulse to provide synchronization with the player's reading of the data input port.

In this mode, input data are generally sampled as described above. However, the Ready pulse is never absent, so it can be used as the indication that a data byte is being read. The leading and trailing edges of this Ready pulse might be used to aid handshaking.

b) The Ready Line is also used to transmit serial data at 9600 baud. This makes it comparatively easy to distinguish the short Ready pulses from any serial data that may appear on the ready line (104 microsecond pulse widths).

3) Serial Data Extraction From Ready

Most UART chips have a false-start-bit detector that requires a valid start bit of at least 8 cycles of the X16 clock (thus 52 microseconds for 9600 baud). Since the Ready read strobe is well below that, a typical UART will ignore the pulse, and capture the serial data, as required, directly from the Ready Line.

4) A Read-Confirm Generated From Ready

A negative-edge triggered single-shot (similar to 74LS123) and an AND-gate can be used to generate a Read-Confirm pulse immediately following a "Ready" pulse.

B. F8 Hardware/Software Monitoring

Through use of the ECI, Pioneer Video, Inc. provides access to the F8 microprocessor's software flags, the software-controlled hardware bits, and the software-monitored hardware status bits.

This information can be obtained from the F8 Registers (not to be confused with user registers), as well as the Input/Output Ports of the F8 microprocessor.

1) F8 Registers

The F8 microprocessor has 64 eight bit registers. Their contents may be of use to applications programmers. Note that any application using these bits becomes implementation dependent.

Register 05	Bit Number	Meaning
	0	Character Generator Disable Flag
	1	Ready as Input Strobe Flag
	2	Process Duplicate Commands Flag
	3	Binary Output Flag
	7	Program Running Flag

Register 06	Bit Number	Meaning
	0 }	Play = 00; Stop = 01;
	1 }	Slow Forward = 10; Slow Reverse = 11.
	2	Freeze Frame Flag
	3	White Flag On Bit
	4	Jump Sync for Slow Motion
	5	Reverse Play Flag
	6	Video Off in Freeze Frame Flag
	7	Door-Was-Opened Flag

Register 08	Contents	Mode Of Player
	*FF	PARK
	*FA-*FE	START-UP Modes
	*F9	PLAY
	*F6	FIELD STEP MODE
	*F8	INPUT
	*F5	AUTOSTOP
	*F4	SEARCH
	*F3	SCAN
	*EF	AUTO SLOW

Register 09	Bit Number	Meaning
	1	Numeric Argument Entry in Progress
	2	Display Off

Registers 46 and 47 contain CLK time.

Register 46 - low order byte
Register 47 - high order byte

Register 61 contains the Landing Pad value

2) F8 Input/Output Ports

The PR-7820 system is monitored by the F8 microprocessor via its Input Ports. In addition, the F8 controls the PR-7820 system modules via the F8 Output Ports.

There are three Input and three Output Ports, each 8 bits wide, which may be accessed via the Transmit Input and Transmit Output Commands. Some of their more useful contents are shown below, and may be better understood by studying the PR-7820 Service Manual.

Input Port	Bit	Meaning
A	5	Test Switch Not Pressed
	6	Door Closed Switch
B	1	Park Position Switch
C	0	In-Focus Flag
	2	Spindle Speed Lock Flag
	4	Outside Limit
	5	Anticipated Outside Limit
	6	Anticipated Inside Limit
	7	Inside Limit

Output Port 0	Bit	Meaning
	1	Closed Loop Tracking
	2	Audio Squelch
	3	Not Video Squelch
	4	Carriage Direction-Not Reverse
	5	Carriage Direction-Not Forward
	6	Carriage Motion - Play
	7	Carriage Motion - Fast

Output Port 1	Bit	Meaning
	0-2	Indicator Light Status: 0-Blank 5-SEARCH 2-STOP 6-SLOW REVERSE 3-PLAY 7-SLOW FORWARD 4-AUTOSTOP
	3	AUDIO 1 ON
	4	AUDIO 2 ON
	5	CAV (not CLV)
	6	Carriage Motion - Slow
	7	Carriage Motion - Medium

Output Port 5	Bit	Meaning
	3	System Squelch
	4	Remote Indicator Off
	6	Lens Down
	7	Spindle Not Turning

6. EXTENDED COMMAND SET

The following paragraphs describe those program commands that facilitate control of the videodisc player by an attached external computer. These commands may also be included in a program dump to allow the player to control an external device, printer, terminal, computer, etc.

A. Enable or Disable Character-Generator and Ready-as-Input-Strobe Flags.

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	EF0	E0		*E0
	EF1	E1		*E1
	EF2	E2		*E2
	EF3	E3		*E3

Function: Two flags are set or reset by the commands EF0 through EF3: the Disable Character-Generator flag, and the Ready-as-Input-Strobe flag.

1) To inhibit player-generated display of numbers and commands, the Disable Character-Generator flag can be turned on. REJECT turns this flag off.

2) Normally, the Ready Line is high when power is on. The line has periodic pulses (16 to 20 microseconds long) that indicate that the player is reading data, and able to execute most commands. These Ready pulses are usually suppressed when the player is executing a program, or in SEARCH, AUTOSTOP, STEP FWD, or STEP REV.

When there are no Ready pulses, the player continues to sample the input, but the user does not know when. The user can enter significant commands at these "non-ready" times to query machine status, to monitor frame numbers and program status, or to trace a running program.

To better enable the interfacing of this data-read port, the player can be instructed to generate the Ready pulse each time data are read, as a read-strobe, even when the player could be considered "Busy." Since the player reads information during the Ready pulse (at least 2 microseconds inside the pulse edges), the user can know whether the data were taken.

CMD	Hex	Ready-as-Input Strobe Flag	Disable Character-Generator Flag
EFØ	*EØ	OFF	OFF
EF1	*E1	OFF	ON
EF2	*E2	ON	OFF
EF3	*E3	ON	ON

Note: These commands cannot be entered when the player is in Automatic Mode.

B. Enable or Disable Binary Output and Unique Input Flags.

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	EF4	E4		*E4
	EF5	E5		*E5
	EF6	E6		*E6
	EF7	E7		*E7

Two flags are set or reset by commands EF4 through EF7: the Binary Output Mode flag, and the Unique Input flag.

1) Transmission from the player is normally one byte of data (although two commands send two bytes of data). These one byte transmissions are usually two ASCII (hex) characters (0, . . ., 9, A, . . . F), with the high order hex character sent first. This technique may prove inefficient for external systems that can accept the full eight bit byte directly. These systems may turn on the Binary Output Mode flag to inhibit the reformatting of the one byte data into two ASCII characters.

The ASCII has the parity bit held at Ø. Thus an ASCII "A" is a hex *41. All bytes are sent low order bit first, using the ASCII convention.

2) Normally, input is sampled once during each field or at about 60 times per second. Since the same input ("button press") could be sampled more than once, the player usually ignores repetitions of the same input on successive reads. Thus, when trying to input 6666 SEARCH, the 6's would all be considered as one "6" if they came one directly after the other. The ENTER line held high, or a *FF command is normally used to separate the 6's into four distinct entries.

An external system, programmed to use the player's data "read strobe", can use it as a "handshake" to know that the player has taken the data. This system can present each command for only one unique read. In this case, if the Unique Input flag is set, data can be presented faster when the sequential commands are identical. Then, the player will consider each input read to be a unique command, even if two or more sequential reads are identical. REJECT turns the flag off.

CMD	Hex	Binary Output Flag	Unique Input Flag
EF4	*E4	OFF	OFF
EF5	*E5	OFF	ON
EF6	*E6	ON	OFF
EF7	*E7	ON	ON

Note: These commands cannot be entered when the player is in Automatic Mode.

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

THIS PAGE
IS
MISSING

Example: The procedure to enter the DROP command (*1D) into location 50 is as follows:

1.	Press	5 0 PROGRAM	Begins Programming Mode, with the screen displaying the contents of location 50.
2.	Press	PLAY	Initiates alternate key entry mode. An "*" is displayed, indicating hex.
3.	Press	1	Generates a 1 - the first of the 2 characters required for the complete definition of the instruction.
4.	Press	SEARCH	Generates a "D" following the "1", thereby completing the DROP instruction entry. The screen address indicator (line 1) is incremented to display location 51. The instruction *1D is displayed and shifted left, out of the three right-most positions of the second display line.
5.	Press	END	Returns the player to Manual Mode.

4. RCU VIEWER RESPONSE

The PR7820-Model-3 programming instruction set includes a variety of instructions which provide a broad range of allowable user input to a program executing in Automatic Mode.

For example, use of the FIN command extends the numeric keypad to provide a numeric range of 0 to 19. Figure 2 illustrates the new values taken on by the already labelled keys when the FIN command is executed. In this manner, the user input keypad becomes a 20 key numeric keypad.

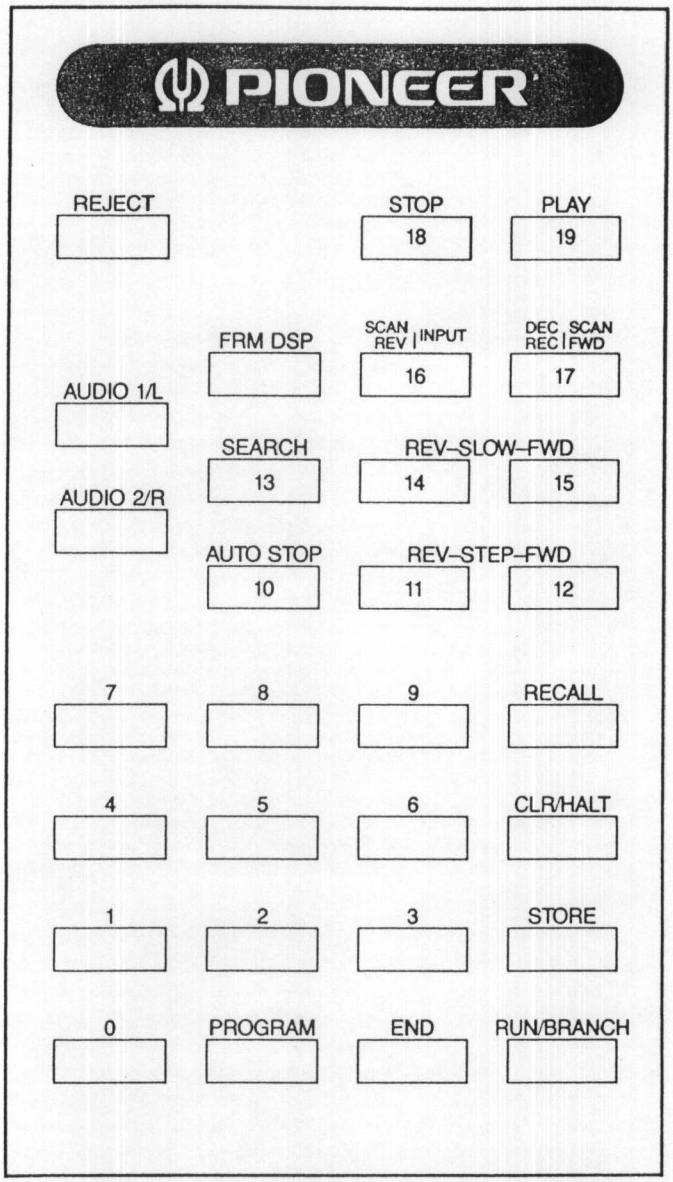


Figure 2. Extended Numeric RCU Pushbuttons

APPENDIX A

SUMMARY OF BASIC COMMANDS

COMMAND[1] ABBREVIATION	CODING MNEMONIC	COMMAND CODE HEX	COMMAND CODE DEC	EQUIVALENT RCU BUTTON	DISPLAYED MNEMONIC	ARGUMENT[2]	TEXT PAGE NO.
ADD	ADD	02	243		*02	Integer	43
ARG	ARG	0A	10		*0A	(Register No.)	44
.AUDIO[1]	A1	F4	244	AUDIO 1	A1	(Integer)	24
.AUDIO[2]	A2	FC	252	AUDIO 2	A2	(Integer)	24
.AUTOSTOP[3]	AS	F3	243	AUTOSTOP	AS	(Frame No.)	27
Axx	AFF	A0	160		*A0		25
"	AFI	A5	165		*A5		25
"	AFN	A1	161		*A1		25
"	AFT	A4	164		*A4		25
"	AIF	AA	170		*AA		25
"	AIN	AB	171		*AB		25
"	AIT	AE	174		*AE		25
"	ANF	A2	162		*A2		25
"	ANI	A7	167		*A7		25
"	ANN	A3	163		*A3		25
"	ANT	A6	166		*A6		25
"	ATF	A8	168		*A8		25
"	ATI	AD	173		*AD		25
"	ATN	A9	169		*A9		25
"	ATT	AC	172		*AC		25
BIN	BIN	17	23		*17		58
.BRANCH	BR	CF	207	BRANCH	BR	(Address)	70
BRF	BRF	07	7		*07	Address	71
CLK	CLK	16	22		*16		45
COM	COM	04	4		*04	Integer	46
.DEC REG	DR	F0	240	DEC REG	DR	Register No.	47
DIN	DIN	1E	30		*1E		59
.DISPLAY	DI	F1	241	FRM DSP	DI	(Integer)	29
DROP[4]	DRP	1D	29		*1D	Integer	49
.END	-	EF	239	END	-		15
FIN	FIN	18	24		*18	Integer	63
FSM	FSM	06	6		*06		30
FTI	FTI	19	25		*19	Integer	65
GET	GET	08	8		*08	Integer	50
.HALT	H	BF	191	HALT	H.		72
.INPUT	IN	F8	248	INPUT	IN	Integer	60
JUMP	JMP	0B	11		*0B	Integer	73
.LOAD	L	CC	204		*CC		19
LPD	LPD	01	1		*01	Integer	32
.NULL	N	00	0		*00		74
.NE	NE	FF	255		*FF		74
PAUSE	PAU	0D	13		*0D	Integer	75
.PLAY	P	FD	253		P.		33
PLOAD	PLD	0C	12		*0C		21
.PROGRAM[4]	-	DF	223	PROGRAM	-	(Address)	14
PUT	PUT	09	9		*09	Register No.	51
.RECALL[5]	RC	7F	127	RECALL	RC	(Register No.)	52
.REJECT	RJ	F9	249		RJ		76
RND	RND	05	5		*05		53

COMMAND[1] ABBREVIATION	CODING MNEMONIC	COMMAND CODE HEX	COMMAND CODE DEC	EQUIVALENT RCU BUTTON	DISPLAYED MNEMONIC	ARGUMENT	TEXT PAGE NO.
.RUN[4]	-	CF	207	RUN	-	(Address)	16
.SCAN FWD[4]	-	F0	240	SCAN FWD			77
.SCAN REV[4]	-	F8	248	SCAN REV	-		77
.SEARCH	SC	F7	247	SEARCH	SC	(Frame No.)	35
.SLOW FWD	SLF	F2	242	SLOW FWD	LF	Frame No.	36
.SLOW REV	SLR	FA	250	SLOW REV	LR	Frame No.	36
.STEP FWD	SF	F6	246	STEP FWD	SF		37
.STEP REV	SR	FE	254	STEP REV	SR		37
.STORE[3]	ST	F5	245	STORE	ST	(Integer)	55
SUB	SUB	03	3		*03	Integer	56
TINPUT	TIN	0E	14		*0E	Integer	67
VOFF	VFF	1C	28		*1C		38
VON	VON	1B	27		*1B		39
.WAIT	W	FB	251	STOP	W.	Integer	40
WFW	WFW	1A	26		*1A		41
.0	0	3F	63	0	0		
.1	1	0F	15	1	1		
.2	2	8F	143	2	2		
.3	3	4F	79	3	3		
.4	4	2F	47	4	4		
.5	5	AF	175	5	5		
.6	6	6F	111	6	6		
.7	7	1F	31	7	7		
.8	8	9F	159	8	8		
.9	9	5F	95	9	9		

Notes: 1 Commands prefaced by a dot, (e.g. .AUDIO 1) are contained in the instruction sets for Model 1 and Model 2 Videodisc Players. Commands without the dot prefix are unique to the Model 3 player.

2 Arguments in parentheses are optional.

3 Command, when executed, activates next register.

4 Not a programmable function

5 Command, when executed, activates next register if register was previously activated by AUTOSTOP, SEARCH, or STORE. If not, command reactivates the active register.

Timing: Each command and each digit of an argument is executed in 1 cycle (1/60th of a second), with the following exceptions:

AUTOSTOP, SLOW FWD/REV - Normal execution time + play duration.

INPUT, TINPUT, FIN, FTI - Normal exec. time + viewer response.

LOAD, PLOAD - 1 cycle, + load duration (up to 7 seconds total).

PAUSE, WAIT - 1 cycle + time specified by programmer.

SCAN FWD/REV - Duration of button press + 12 cycles.

SEARCH - Normal execution time + search duration.

STEP FWD - 3 cycles.

STEP REV - 4 cycles.

WFW - 1 or 2 cycles.

APPENDIX B

SUMMARY OF COMPUTER COMMANDS

CODING MNEMONIC	COMMAND CODE HEX	COMMAND CODE DEC	DISPLAYED MNEMONIC	ARGUMENT	TEXT PAGE NO.
Enable or Disable Binary Output and Unique Input					
EF4	E4	228	*E4		89
EF5	E5	229	*E5		89
EF6	E6	230	*E6		89
EF7	E7	231	*E7		89
Enable or Disable Character Generator and Ready Flags					
EF0	E0	224	*E0		88
EF1	E1	225	*E1		88
EF2	E2	226	*E2		88
EF3	E3	227	*E3		88
Increment or Decrement Transmit Pointer					
DTP	DA	218	*DA		94
ITP	D9	217	*D9		94
Set Transmit Pointer					
STP	E8	232	*E8	Integer	93
Transmit Contents of F8 Input Ports					
TIA	D5	213	*D5		98
TIB	D6	214	*D6		98
TIC	D7	215	*D7		98
Transmit Contents of F7 Output Ports					
TP0	D0	208	*D0		99
TP1	D1	209	*D1		99
TP5	D2	210	*D2		99
Transmit Contents of RAM					
TM	DC	220	*DC		95
ITM	DD	221	*DD		95
DTM	DE	222	*DE		95
Transmit Frame Number					
TFN	D3	211	*D3		91
Transmit F8 Register					
TR	D8	216	*D8		97
Transmit Program Address Pointer					
TPA	DB	219	*DB		100
Transmit Status					
TS	D4	212	*D4		92

Note: The commands listed in this table are not contained in instruction sets for the Model 1 and 2 Videodisc Players.

Each command and each argument digit is executed in 1 cycle (1/60th of a second).

USER'S COMMENT FORM

Photocopy this page as often as needed, complete the copy and mail to:

Pioneer Video, Inc.
Customer Program Development Services
3300 Hyland Avenue
Costa Mesa, California 92626

Please use this form to comment on this publication, suggest additions or deletions, and list specific errors and omissions (give page numbers).

Pioneer Video, Inc. may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

If you wish a reply, be sure to include your name and address.

COMMENTS